Natural Language Processing Journal 8 (2024) 100101

Contents lists available at ScienceDirect

Natural Language Processing Journal

journal homepage: www.elsevier.com/locate/nlp

Towards effective teaching assistants: From intent-based chatbots to LLM-powered teaching assistants

Bashaer Alsafari [a,c,*], Eric Atwell [a], Aisha Walker [b], Martin Callaghan [a]

[a] School of Computing, University of Leeds, Leeds, United Kingdom
[b] School of Education, University of Leeds, Leeds, United Kingdom
[c] Applied College, King Abdulaziz University, Jeddah, Kingdom of Saudi Arabia

ARTICLE INFO

Keywords:
Teaching assistant chatbot
Intent-based chatbot
Large language models (LLMs)
Retrieval-augmented generation (RAG)

ABSTRACT

As chatbot technology undergoes a transformative phase in the era of artificial intelligence (AI), the integration of advanced AI models emerges as a focal point for reshaping conversational agents within the education sector. This paper explores the evolution of educational chatbot development, specifically focusing on building a teaching assistant for Data Mining and Text Analytics courses at the University of Leeds. The primary objective is to investigate and compare traditional intent-based chatbot approaches with the advanced retrieval-augmented generation (RAG) method, aiming to improve the efficiency and adaptability of teaching assistants in higher education. The study begins with the development of an Amazon Alexa teaching skill, assessing the efficacy of traditional chatbot development in higher education. To enrich the chatbot knowledge base, the research then employs an automated question–answer generation (QAG) approach using the QG Lumos Learning tool to extract contextually grounded question–answer datasets from course materials. Subsequently, the RAG-based system is proposed, leveraging LangChain with the OpenAI GPT-3.5 Turbo model. Findings highlight limitations in intent-based approaches, emphasising the need for more adaptive solutions. The proposed RAG-based teaching assistant demonstrates significant improvements in efficiently handling diverse queries, representing a paradigm shift in educational chatbot capabilities. These findings provide an in-depth understanding of the development phase, specifically illustrating the impact on chatbot performance by contrasting traditional methods with large language model-based approaches. The study contributes valuable perspectives on enhancing adaptability and effectiveness in AI-powered educational tools, providing essential considerations for future developments in the field.

1. Introduction

The rapid growth of digital technologies in the field of education has led to the rise of alternative learning models in higher education institutions. In technology-enhanced learning systems, the learning models have evolved from e-learning and mobile learning to smart learning environments, which utilise artificial intelligence (AI) and modern technologies to offer students more personalised, flexible, and motivating learning experiences (Spector, 2014) . Engaging and supporting students in the learning process is crucial for improving their focus and maintaining their motivation. Several studies have shown that the lack of individual support from instructors leads to poor learning outcomes and high student dissatisfaction (Eom and Ashill, 2016) . However, providing individual support in higher education has always been a costly and challenging task for educators and academic departments. To tackle this issue, intelligent assistant agents known as chatbots can be used to provide students with personalised support at any time of the day.

Chatbots are computer programs that imitate human conversation using natural language (Fryer et al., 2017). This technology is also referred to as a conversational agent, conversational interface, dialogue system, virtual assistant, and personal assistant (Altinok, 2018). The primary role of a chatbot is to comprehend user inputs and provide intelligent responses through text or voice-based interaction. In higher education, teaching assistant chatbots can simulate the challenging educational task of teaching. In this context, teaching assistants offer an automated way to support students individually, accurately answer their questions, and provide personalised responses. The effectiveness of teaching assistant chatbots heavily relies on the methodology used in their development, particularly in how they interact with the underlying learning materials. While the idea of using chatbots in education is not new, the recent advancement of generative AI has ushered in a new era in their development process. Therefore, it is important to ensure that studies and evaluations reflect the latest advancements and developments in the field.

* Corresponding author at: School of Computing, University of Leeds, Leeds, United Kingdom.
E-mail address: scbha@leeds.ac.uk (B. Alsafari).

https://doi.org/10.1016/j.nlp.2024.100101
Received 6 April 2024; Received in revised form 2 August 2024; Accepted 20 August 2024

Traditionally, the development of chatbots has commonly relied on intent-based approaches, particularly before the release of more advanced generative AI models like generative pretrained transformer (GPT). Intent-based chatbots work by detecting the user's intention and responding with predefined responses and patterns. These chatbots typically use sophisticated natural language understanding (NLU) algorithms and machine learning (ML) techniques to analyse user inputs and classify their intentions. As a result of this analysis and intent classification, the chatbot can be trained to respond to user requests with predefined responses. However, developing and maintaining intent-based systems requires significant resources, as the developer must determine all possible user intentions, how the chatbot should respond, and any functions or data the system may need (Misargopoulos et al., 2022). Additionally, in the context of domain-specific chatbots, there is a fundamental need for a well-defined and high-quality dataset of questions and answers to enhance and train the system. Constructing such datasets presents significant challenges that require considerable curation and domain expertise.

Nowadays, large language model (LLM)-based chatbots offer a promising solution to overcome these limitations, utilising their advanced natural language processing (NLP) capabilities to create more adaptable and powerful conversational interactions. LLMs are sophisticated AI models that can perform various NLP tasks, such as text generation and conversational question answering, by analysing and learning patterns from extensive amounts of textual data. To develop a chatbot specific to a particular domain, foundational models can be enhanced by integrating contextual material through the process known as retrieval-augmented generation (RAG) (Lewis et al., 2020). RAG is a technique for enriching LLM knowledge with additional domain-specific data. It is a robust approach to constructing domain-specific knowledge chatbots that can extract information and generate responses from unstructured data, including audio and video transcripts, PDFs, Word documents, and more. This approach empowers developers to create custom-knowledge chatbots powered by generative AI capabilities.

To empirically evaluate the effectiveness of traditional intent-based development compared to the more advanced approach of RAG-based systems with LLM models in the education sector, we conducted a comparative research study. Our main focus was on creating a chatbot for teaching assistants specifically for the Data Mining and Text Analytics courses within computing programs at the University of Leeds. The primary goal of this chatbot is to provide students with round-the-clock support by answering their questions and clarifying any uncertainties they may have about the course content. The chatbot acts as a personalised teaching assistant, available at all times to offer support and guidance to each student. The study starts with building a teaching skill for Amazon Alexa, evaluating the practicality of developing traditional intent-based chatbots for higher education. To enhance the chatbot's knowledge base, the research then utilised an automated question–answer generation (QAG) approach using the QG Lumos learning[1] tool to extract contextually grounded question–answer (QA) datasets from course materials. Subsequently, the RAG-based system was proposed, utilising LangChain with the OpenAI GPT-3.5 Turbo model to evaluate the impact of generative AI approaches in teaching. The following research questions (RQs) guided our exploration:

RQ1: How do the development processes differ between traditional intent-based chatbots and RAG-based systems when using course materials to build a teaching assistant chatbot?

RQ2: How do LLM-based chatbots, particularly those utilising RAG, enhance the effectiveness of teaching assistant chatbots in higher education?

The effectiveness of a teaching assistant chatbot in the educational domain depends on its ability to provide responses that are relevant, correct, accurate, clear, and insightful. The chatbot should be able to answer students' questions with relevant and factually correct responses. Additionally, the provided information should be clear, easy to understand, and offer a deep understanding of the educational context. Overall, the chatbot's interactions should be helpful and effectively support the students' learning process. This research will evaluate how different approaches can meet these criteria, specifically focusing on relevance, correctness, accuracy, clarity, depth of thought, and overall helpfulness.

This study contributes to the expanding field of AI in education by providing valuable insights into how different methodologies impact the effectiveness of teaching assistant chatbots. The findings have practical implications for future developments in the design of educational chatbots, aiming to inform and guide best practices in the field.

The remainder of this paper is structured as follows. Section 2 presents the literature review. Next, in Section 3, we introduce the data source utilised in our experiments. Section 4 outlines the methodology used for our empirical exploration. Then, in Section 5, we present each experiment and analyse the challenges and results. Section 6 discusses the practical implications of our findings for the design and implementation of teaching assistant chatbots in higher education. Finally, in Section 7, we draw conclusions from our research and offer insights for future studies aiming to enhance the capabilities of educational chatbots.

2. Literature review

2.1. Chatbots as teaching assistant agents

Our research focuses on the development of chatbot technologies in the field of education, recognising their potential to significantly enhance higher-education environments. For instance, a chatbot can be available around the clock to support and answer student questions regarding admission, registration, grades and courses. Several studies have investigated the use of chatbots in teaching, learning, and support (Hien et al., 2018; Hobert, 2019; Jia, 2009; Shawar and Atwell, 2007). Building chatbots as teaching assistant agents is more complex than completing tasks or answering FAQs. In teaching, chatbots can assist teachers in answering domain-specific questions, tracking student learning progress, and providing personalised feedback (Clarizia et al., 2018). In other words, a teaching assistant chatbot can provide information to students on a specific topic similar to a human tutor, thereby relieving teacher workloads (Perez et al., 2020). From the students' perspective – especially first-year students (Carayannopoulos, 2018) – the chatbot can provide an interactive learning experience (Clarizia et al., 2018), that helps them manage the information load and feel more socially engaged. Furthermore, in online courses, building interactive communication with students is crucial for keeping them motivated and engaged (Song et al., 2017). Therefore, using a chatbot as a teaching assistant in face-to-face or online courses can improve both the learning environment and student performance.

Several studies have explored the utilisation of chatbots as teaching assistant agents. For example, Code Tutor (Hobert, 2019) stands out as a chatbot-based learning system crafted to aid students in introductory programming courses. This chatbot assists students in mastering software code writing by addressing open-ended questions, furnishing automatic assessment results, and leading them through coding exercises step by step, employing natural language interaction. Similarly, Python-Bot (Okonkwo and Ade-Ibijola, 2020), focuses on imparting basic Python syntax and semantics to students. Another noteworthy example is the Mobile chatbot (Pham et al., 2018) designed as a language-learning tool that motivates learners through various linguistic exercises and gamification techniques. Additionally, CSIEC (Jia, 2009) and Bookbuddy (Ruan et al., 2019) represent chatbot systems supporting students in English language learning across different educational levels. With the proliferation of educational materials available

[1] https://www.lumoslearning.com/llwp/free-question-answer-generator-online.html

to students, including textbooks, lecture slides, exercises, and scientific papers, these studies exemplify practical methods of leveraging chatbots to support and navigate students through the learning process. While the current development of chatbots in the education sector has yielded promising results, the emergence of LLM-based chatbots signals a paradigm shift in traditional chatbot development. Therefore, further investigation is crucial to align with the evolving landscape of LLM technology, particularly in the field of education.

2.2. Intent-based chatbot

Traditional approaches in conversational user interfaces, such as virtual assistants like Alexa, have typically focused on intent-based chatbots. These platforms, operating at various levels, rely on intent-based systems supported by NLP, ML, and deep learning (DL) techniques, providing user-friendly frameworks for developing chatbot systems. They are built upon a foundation that incorporates rule-based systems for recognising intents and entities. However, to enhance their understanding of natural language and improve their adaptability, they integrate advanced technologies such as NLP, ML, and DL techniques. Recently, LLMs have emerged as a novel approach to address specific limitations observed in intent-based conversational systems.

Intent-based chatbots are designed to identify and analyse user intentions using predefined patterns and commands (Luo et al., 2022). They primarily rely on decision trees and structured knowledge bases, which enable them to respond to specific requests and perform tasks based on recognised statements. Despite their limitations in flexibility and adaptability, this method ensures a substantial level of control and predictability in interactions, making intent-based systems well-suited for applications prioritising precision and rule adherence. Nevertheless, the inherent inflexibility of intent-based systems poses significant challenges (Rahman et al., 2017). These challenges involve difficulties in understanding subtle expressions and functioning efficiently in dynamic contexts with rapidly changing user needs. For example, in education, an intent-based chatbot designed to assist students with course-related queries may struggle to comprehend complex questions and adapt to changes in the curriculum or learning materials. Previous studies highlight how intent-based chatbots often fall short in assisting users in accomplishing their goals (Folstad et al., 2021; Lee et al., 2021; Meyer vonWolff et al., 2021).

In education, intent-based systems have been explored for facilitating various educational tasks. For instance, Chien and Yao (2020) developed an AI userbot system that combines intent-and flow-based dialogue modes to enable engineering students to interact with virtual product users, facilitating participatory design activities. EDUBOT, a chatbot designed to assist students during the COVID-19 pandemic by addressing academic queries through predefined intents, helps students learn their subjects through a question-and-answer format (Sophia and Jacob, 2021). Similarly, HSchatbot is a chatbot intent classifier that supports high school students by predicting the intent of their enquiries related to academic choices such as scholarships, university requirements, majors, and curriculums (Assayed et al., 2023). Additionally, Wang et al. (2022) developed an educational chatbot that utilises joint intent classification and slot filling models aimed at enhancing online learning experiences by understanding task-oriented natural language texts to provide education-related services.

Most of these systems rely on predefined intents and predetermined scripts, aiming to improve user understanding and intent recognition. However, in educational settings, there is a greater need for flexible and contextually rich knowledge support, which intent-based systems often fail to provide due to their inherent design limitations. In this study, the intent-based model serves as a benchmark for comparison with more advanced models, such as LLM-based chatbots, due to its traditional, well-known, and widely adopted framework. Intent-based systems represent a well-established and extensively studied technology in the chatbot domain. Their established use in various educational applications makes them an ideal foundation for evaluating the potential enhancements offered by more sophisticated LLM-based models. It is essential to explore LLM-based chatbots to assess how these advanced models can more effectively adapt to and address the evolving needs of the educational sector.

2.3. LLM-based chatbot

Most recently, advanced generative AI chatbots utilising LLMs have revolutionised the world with their remarkable capabilities. These conversational AI systems are pretrained on vast amounts of data and leverage DL and NLP techniques to generate human-like responses. A major advancement in generative AI has been made with the launch of ChatGPT[2] and many other LLMs such as LaMDA (Thoppilan et al., 2022), GPT-NeoX (Black et al., 2022), PaLM 2 (Anil et al., 2023) and LLaMA 2 (Touvron et al., 2023). They excel at understanding, generating, and interacting with human language (Jawahar et al., 2019), leading to more sophisticated and context-aware responses compared to the rigid structures of intent-based systems. Moreover, LLMs represent a significant improvement over previous language processing approaches, including rule-based systems and recurrent neural networks (RNNs). Rule-based systems lack flexibility in handling diverse language patterns, while RNNs struggle with long-range dependencies in sequential processing (Luo et al., 2022). LLMs can simultaneously generate and analyse text, effectively managing extensive contextual information and complex linguistic patterns (Zhao et al., 2023). OpenAI,[3] a well-known participant in this field, has made significant contributions to this advancement through its series of GPT models. In a wide range of NLP applications, such as text generation, question-answering, sentiment analysis, and language translation, GPT models have achieved state-of-the-art performance (Ray, 2023). One of the main benefits of using GPT for chatbot development is its ability to generate coherent and contextually appropriate text that closely resembles human natural language. GPT is a versatile, valuable, and multipurpose tool that can be customised for specific tasks or datasets. This adaptability allows for the creation of effective chatbots that can handle a wide range of interactions (El Alaoui et al., 2023). However, their accuracy may not always be perfect.

In educational settings, chatbots depend heavily on unstructured data sources like textbooks, slides, lecture transcripts, and research papers. This poses a significant challenge for NLP. State-of-the-art LLMs provide a solution to address these challenges in educational chatbots using methods like RAG (Lewis et al., 2020). This approach allows for the efficient extraction of relevant information from unstructured documents when responding to user queries. RAG-based systems possess the capability to address the inherent limitations of intent-based models. Intent-based chatbots rely on predefined intents and responses, which restrict their flexibility and contextual comprehension. In contrast, RAG-based systems dynamically retrieve information from extensive corpora of documents to generate customised responses tailored to the user's specific requests (Maryamah et al., 2024). This approach results in responses that are more accurate and contextually enriched. RAG employs integrated retrieval and generation techniques that collaboratively enhance contextual understanding. The retrieval component identifies the top k text passages that are relevant to the input query, thereby improving the model's comprehension and the generation of responses. This process is represented by the equation $p_n(z|x)$, where p_n denotes the retrieval component with parameters n (the number of documents or passages to be retrieved), selecting relevant passages z from the vector databases based on the input x (Neupane et al., 2024). This methodology is particularly beneficial in educational settings, where the diversity and complexity of student queries often surpass the capabilities of intent-based systems. The RAG approach provides a more flexible and scalable solution, adept at handling a broad range of topics

[2] https://chat.openai.com

[3] https://platform.openai.com/docs/models

and complex questions typical in educational contexts. For instance, the BARKPLUG V.2 system at Mississippi State University demonstrated superior performance in generating accurate and relevant responses to domain-specific questions by utilising the RAG approach, significantly enhancing user satisfaction and engagement (Neupane et al., 2024). Similarly, the integration of AI tools in teaching an introductory course in computer science at Harvard University demonstrated the effectiveness of RAG in providing detailed and contextually appropriate responses, which facilitated personalised tutoring for students (Liu et al., 2024)

The use of LLM-based chatbots in education is still in its early stages, with a limited number of papers conducting experiments and evaluating their capabilities within the field (Khadija et al., 2023; Soygazi and Oguz, 2023). Thus, this study contributes to the existing literature by investigating the effectiveness of LLMs in developing a teaching assistant chatbot for higher education.

3. Data source

This research is based on course materials from the Data Mining and Text Analytics course, including textbook chapters and lecture transcripts (see Appendix for topics covered in the lectures). The aim is to provide assistance to computing students who are enrolled in the online master's course at the Computing School of the University of Leeds. The chatbot is designed to simulate the role of a teaching assistant, offering support and answering questions based on the provided course content. The foundational data for developing a domain-specific chatbot, which serves as a teaching assistant, is extracted from these course materials. The data is presented in PDF format, with file lengths ranging from 5 to 40 pages. Approximately 500 pages of course-related content were processed, which encompass various elements such as text, equations, tables, and figures.

4. Methodology

The methodology used in this study outlines a systematic sequence of experiments designed to comprehensively develop and evaluate both traditional intent-based chatbots and LLM-based chatbots, with a particular focus on enhancing the development of educational chatbots for higher education teaching. The main goal is to create an advanced teaching assistant that helps deliver educational materials by answering students' questions and providing on-demand access to course information. The process involves a series of experiments aimed at ensuring the effectiveness, accuracy, and relevance of the implemented chatbot systems. This section provides a detailed exploration of each experiment in the proposed sequence, presenting the main objectives, inputs, and outputs.

For intent-based chatbot development, the research begins by developing an Amazon Alexa teaching skill using a traditional chatbot development framework, as described in Section 5.1. The initial focus is on investigating the adoptability of an intent-based chatbot schema for building an effective, scalable, and reliable teaching assistant chatbot. The design and implementation of the intent-based chatbot involved creating a customised Alexa skill with four main components: Invocation (specific phrases to initiate interaction), Intents (models for actions based on user requests), Sample Utterances (phrases or questions to invoke intents), and Slots (variables embedded in utterances). To enhance Alexa's question recognition, various example phrases were utilised for each question, and the knowledge base, containing appropriate responses, was stored as a JSON file. The performance of the intent-based chatbot relies mainly on the quality of the chatbot knowledge base. Thus, the primary challenge was enriching the chatbot knowledge base with an extensive training dataset (QA dataset) generated from unstructured data such as textbooks and lecture transcripts. Manual extraction is a complex and time-consuming process, which led to exploring automated QAG techniques.

In the next stage of the study, the goal was to generate QAs. The objective was to automatically create a high-quality QA dataset relevant to the context, using teaching materials from the course such as textbooks and lecture transcripts. To accomplish this, an automated QAG web tool called Lumos Learning QG was used, which overcame the limitations of the manual method. The generation methodology involved preprocessing the PDF content by removing graphs, tables, equations, file headers, and footers, and extracting information using the Tika library in Python. The cleaned text was then split into well-formed paragraphs and submitted to the Lumos Learning QG tool, and the generated QAs were exported to Excel files. The quality of the QA pairs was evaluated through several metrics including correctness, relevance, accuracy, clarity, depth of thought, and overall helpfulness. This part of the process is explained in Section 5.1.3.

The findings from previous experiments, which indicated that the intent-based framework, along with generating a dataset of questions and answers from unstructured learning materials, was not sufficiently effective in reflecting the persona of a teaching assistant chatbot, prompted the researchers to compare it with the latest evolution of LLM-based chatbots. This involved using the RAG approach to build a teaching assistant chatbot. The methodology employed here utilises LangChain with the OpenAI GPT-3.5 turbo model, combining the power of retrieval and generation mechanisms to create a custom knowledge teaching chatbot for higher education. The design and implementation of the LLM-based chatbot involved preprocessing educational data, generating embeddings, and using vector retrieval to retrieve and generate relevant responses. The detailed development process of the RAG-based chatbot is explained in Section 5.2. This evolution aims to address the limitations identified in the earlier phase of the research, enhancing the capabilities and effectiveness of the educational chatbot. The following section will discuss the methodology, results, and evaluation of each experiment.

5. Experimental setup and design

5.1. Educational intent-based chatbot development (Amazon Alexa skill)

Traditional intent-based chatbots operate with the goal of understanding and fulfilling user intentions within a specific context. These chatbots are designed to comprehend user queries, whether spoken or written, and provide responses that align with the intended request. Typically, intent-based chatbots utilise advanced NLU algorithms to analyse user inputs into predetermined semantic slots (Chen et al., 2017) and classify user intentions using ML techniques (Franco et al., 2020). The generation of responses in intent-based chatbots heavily relies on knowledge bases that include intentions, training phrases, and referenced responses (Rosruen and Samanchuen, 2018). Fig. 1 illustrates the general operational flow of a traditional intent-based chatbot.

For example, consider the following student inquiry seeking assistance with programming homework:

```
User: I'm struggling with my programming homework. It is
an assignment on implementing data structures in Python,
specifically a linked list. Can you help me understand how
to approach this?
```

In this request, the system should discern the user's intent, which is to 'seek help with programming homework', with a specific focus on 'implementing a linked list in Python'. Furthermore, entities within the user input should be identified to extract contextually relevant details as follows:

```
Homework Type Entity: "programming homework"
Programming Language Entity: "Python"
Data Structure Entity: "linked list"
```

This identification of intent and entities is a critical aspect of intent-based chatbots. It enables the system to fully understand the user's

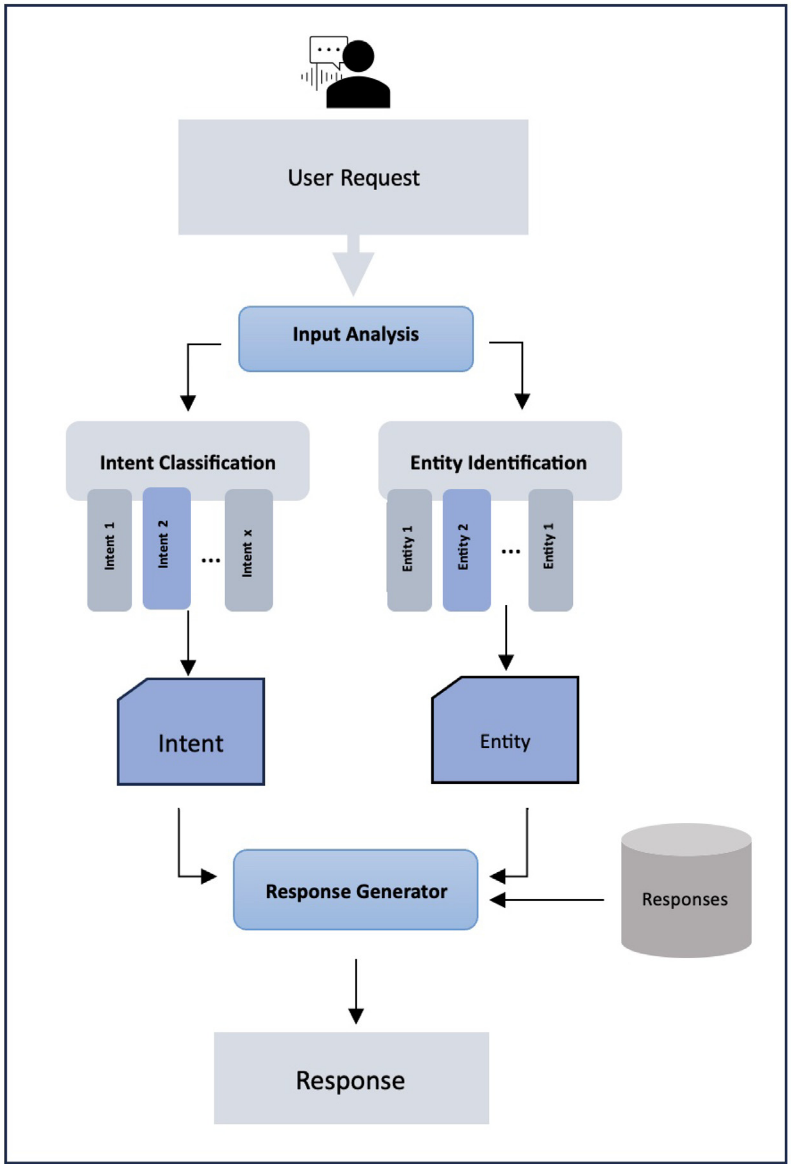

Fig. 1. Intent-based framework.

request and provide targeted support and guidance tailored to the specific user needs. By identifying key entities, the chatbot ensures that its responses are not only contextually relevant but also accurately aligned with the user's inquiry, thereby enhancing the overall effectiveness of the conversational experience. To explore the capabilities of intent-based chatbots in an educational context, we selected Amazon Alexa as the platform for our study. Amazon Alexa makes significant contributions to the development of intent-based chatbots. Its user-friendly interface and extensive skill development opportunities make it an ideal environment for investigating the effectiveness and educational potential of intent-based chatbots in higher education. Our primary objective was to develop an Alexa skill that simulates a tutor assistant for the Data Mining and Text Analytics course, aiming to uncover the potentials, challenges, and limitations inherent in this approach.

Alexa, a cloud-based voice assistant service developed by Amazon for its Echo, Echo Dot, Echo Show, and Echo Studio devices (Alexa, 2023c), offers voice interaction, real-time information, weather forecasting, and other services. Developers can use Amazon's Alexa Skill Kit (ASK) to create and customise custom 'skills' for Alexa. These skills function like applications, allowing developers to define specific tasks and perform them through an interactive voice interface. The following section describes the system design and methodology, followed by the evaluation of the chatbot.

5.1.1. Design and implementation

We have created a customised Alexa skill to examine the possibilities and difficulties of incorporating Alexa into university education. The skill is made up of four main components.

- Invocation: This term refers to a specific phrase spoken by the user to initiate the interaction, such as 'Hi Alexa, open the data mining skill'.
- Intents: These function as structural models for actions that fulfil a user's spoken request when an intent utterance is recognised. Table 1 provides a summary of the developed intents.
- Sample utterances: These are lists of phrases that users can say to invoke a particular intent.
- Slots: These are variables that can be optionally parsed and embedded in the sample utterances to activate a skill.

Depending on the type of intent and the value of the slot, Alexa searches for the most suitable answer in the knowledge base. Once an

Table 1
Summary of skill intents.

	Intent name	Description
1	TermsIntent	Answer students' questions about the main course terminologies by providing their definitions from course materials.
2	HowToIntent	Capture students' requests on how a computational function/concept works.
3	ExamplesIntent	Provide students with examples of course concepts.
4	GeneralIntent	Address other general questions about the course content not specified by other intents.

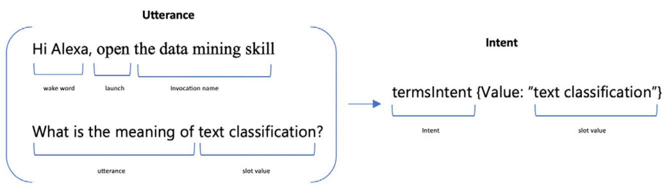

Fig. 2. Question processing mechanism within the termsIntent.

answer is found, a response is generated using the Lambda function, which is an essential component of AWS Lambda responsible for executing the backend logic of the skill (Alexa, 2023b). This serverless function processes the user's request, identifies the appropriate intent, and triggers the corresponding actions and responses, ensuring smooth communication between the Alexa skill and its backend.

For the purposes of the experiment, important elements of the course were taken from the textbook and lecture transcripts to create the knowledge base. Four main intents with 20 QA pairs were set up: termsIntent, howToIntent, examplesIntent, and generalIntent. A summary of the developed intents is shown in Table 1. To improve Alexa's ability to recognise questions, different example phrases were used for each question. The chatbot's knowledge base, containing the appropriate responses for each question, is stored as a JSON file.

In Fig. 2, we provide an illustrative example that highlights the question processing mechanism of the termsIntent, which follows the Amazon Alexa design model (Alexa, 2024). This intent is set up to handle enquiries concerning the meanings or definitions of important terms used in the course. For example, if a user asks about the term 'text classification', the system identifies it as an instance of the 'termsIntent' and extracts the specific term, such as 'text classification', as a slot value associated with the term entity. Based on the recognised intent and entity, the system retrieves information from the knowledge base to provide precise and contextually appropriate responses.

Finally, the interaction model was defined and implemented using ASK. ASK handles the user's voice-based request as an audio signal. The audio input is then converted into a text message, which activates the identified intention based on the user's request. When the intention is invoked, the corresponding action and response are executed using the Lambda function. Subsequently, Alexa converts the generated responses into an audio output for the user. Fig. 3 shows the diagram of the custom Alexa skill (Alexa, 2023a) .

5.1.2. Testing and evaluating

The Alexa skills that were developed underwent testing with the invoking phrase and several questions. Before deploying the skill, the specified utterances were tested using the 'utterance profile' tool. This systematic step aimed to confirm that the skill could accurately resolve utterances to intents and slots. For example, during the testing phase, similar utterances such as 'Define machine learning' and 'Explain the concept of machine learning' were included to ensure that the skill effectively captured the variations in user queries related to the same intent. This approach helped validate the skill's robustness in recognising diverse expressions of user intent.

Simultaneously, to test the overall functionality of the skill, we used the ASK Alexa Simulator tool. This tool provides an interface for text and voice interactions that allows us to invoke the developed skill as if the user were using an actual device. This included both text and voice interactions, effectively replicating real-world user engagement scenarios. During the test, we asked 10 spoken questions and 10 equivalent written questions, using different phrasing and introducing new terms. For the spoken test, Alexa successfully handled and answered three of them correctly. For the written questions, Alexa appropriately replied to five of them. However, if the question was formulated differently or included new words that were not part of the sampled utterances, Alexa did not recognise the user's request and replied with a default response. In some cases where the questions were unrecognised or unexpected, Alexa responded with suggested answers extracted from the web. Fig. 4 illustrates a user interaction with the Amazon Alexa teaching skill. When the user posed a question, "Can you explain the concept of lemmatisation', Alexa failed to identify the intended intent and slot. However, when the user phrased the question more simply as, 'What is the meaning of lemmatisation', it responded correctly. This observation indicates that Alexa may encounter challenges when faced with questions diverging from the specific language patterns it was trained on. Consequently, users may need to conform to these patterns for optimal system performance, indicating a potential constraint in addressing a wider range of diverse user queries.

Overall, Alexa is a user-friendly platform that simplifies the development process of chatbots and provides a model for voice interactions. However, all skill data must be generated and entered manually. This may be efficient for small, simple, and predictable scenarios, but it is time-consuming for larger and more complex ones. Furthermore, a system that relies on a predefined knowledge base requires a large dataset of question-and-answer pairs for continuous improvement. This dataset plays a crucial role in enhancing the system's ability to identify various user enquiries and cover a wider range of information. However, manually collecting and generating a QA dataset from course learning materials proves to be a complex and time-consuming process. Therefore, the following section focuses on expanding the knowledge base through automatic question–answer pair generation techniques. This approach aims to refine the chatbot's knowledge base using course materials, overcoming manual limitations, and improving the chatbot's proficiency in handling diverse educational queries.

5.1.3. Automated question–answer pair generation for chatbot knowledge base expansion

In order to address the limitations identified in the initial experiment, the second phase of the research was conducted to improve the chatbot's knowledge base. Typically, domain-specific chatbots heavily rely on a structured knowledge base that consists of QA pairs. These QA pairs are used as the main resource for answering user enquiries, by determining the semantic similarity between the user's query and the predefined questions in the knowledge base. In this approach, the collection or generation of QA pairs is a critical task in providing informative and relevant responses. Constructing QA pairs from educational materials, such as textbooks and lecture transcripts, requires efficiently determining the desired question type and its corresponding answer. When generating question–answer pairs, quality is an important factor, especially in education. High-quality questions must be framed in clear, concise, meaningful, and grammatically correct language. Furthermore, the corresponding answer should exhibit the same characteristics and be comprehensively correlated to the question. Moreover, in teaching, it is essential that the knowledge base covers the entirety of educational content to accommodate diverse question types.

Generating such QA pairs manually is both expensive and time-consuming. Therefore, several studies have explored models and techniques for automatically generating QA pairs from unstructured text (Aithal et al., 2021; Cui et al., 2021; Willis et al., 2019). QAG is the process of simultaneously applying question generation (QG) and answer

Fig. 3. Alexa custom skill diagram. (Alexa, 2023a).

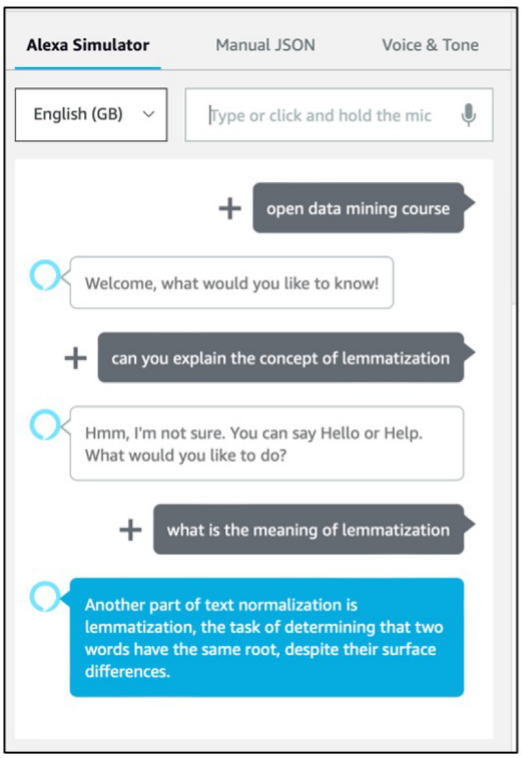

Fig. 4. Example of Alexa responses.

generation (AG) techniques to a given passage. This approach involves generating a set of questions specifically related to the content of the passage and providing contextually relevant answers to each of these questions. QAG models aim to generate a large set of QA pairs from natural language texts, thus addressing the challenges associated with manual QA generation. To evaluate the effectiveness of this approach, we conducted an experiment using the Lumos Learning QG web tool to automatically generate question–answer pairs for a teaching assistant chatbot.

Question–answer generation methodology. Initially, we identified the available, free, and useful web tools for generating questions. In Al-nefaie et al. (2023), the authors evaluated these web services based on various quality criteria such as the clarity, syntax, and meaning of the generated questions and answers. Their evaluation revealed that the Cathoven QG, ExploreAI QG Question Generation demo, and Lumos Learning QG tools achieved the highest scores across all criteria. However, during the experiment, the ExploreAI QG demo was unavailable, and the Cathoven QG website was unstable. Therefore, the experiment began by examining the Lumos Learning QG tool for generating educational QA pairs from the course learning materials. Lumos Learning QG is a web-based solution that uses advanced AI and ML algorithms to generate questions and answers from text. Users can paste a paragraph with a minimum of 500 characters into the tool, and the system generates a set of questions and answers related to the submitted text. The generated QA pairs can be exported in CSV format.

The Lumos Learning QG tool is specifically designed to handle textual information. However, when dealing with scientific educational content, such as textbooks, which frequently include a combination of graphs, tables, headers, and footers on each page, a preprocessing step was necessary to enhance the extraction of information from PDF files. The QA generation process followed this procedure:

- Cleaning the PDF content by removing graphs, tables, equations, file headers, and footers.
- Extracting information from PDF files using the Tika library in Python. Tika is a library used for document type detection and content extraction from various file formats, including PDF.
- Following the extraction, the text was split into paragraphs of the required length with well-formed sentences.
- Submitting the prepared text to the Lumos Learning QG tool.
- Exporting the generated QAs to Excel files.

Result and evaluation. The Lumos Learning QG tool produced a total of 9000 QA pairs, with 3570 sourced from the textbook chapters and 5430 from the lecture transcripts. The evaluation aimed to assess the quality of the generated QA pairs for the development of educational chatbots. A random sample of 200 QA pairs was chosen for detailed evaluation by two annotators with expertise in data mining and text analytics. Annotators used a scale ranging from 1 to 5 to independently rate each metric for each QA pair, providing a comprehensive evaluation of the generated content. The metrics used in the evaluation included correctness, relevance, accuracy, clarity, depth of thought, and overall helpfulness, offering a comprehensive assessment of the QA pairs. Each QA pair was rated on a scale from 1 to 5, with 1 indicating the lowest score and 5 indicating the highest score.

The correctness metric evaluates whether the question and answer together form a meaningful and coherent pair. Relevance assesses the alignment between the generated answer and the content of the question in relation to the passage. Accuracy assesses the precision of information in both the question and the answer. Clarity evaluates the presentation of both the question and answer in terms of language and structure. Depth of thought measures the depth to which both the question and the answer reflect critical analysis and understanding of the topic. Overall helpful evaluates the practical usefulness of both the question and the answer as a pair, considering their value and applicability, especially in educational contexts.

As shown in Table 2, the metrics for correctness and relevance indicate a moderate level of satisfaction, with scores of 3.62 and 4.01, respectively. This suggests that, on average, the generated pairs form meaningful and contextually aligned combinations. However, challenges arise in terms of accuracy, clarity, depth of thought, and overall helpfulness, with scores ranging from 2.82 to 1.84. The accuracy score of 2.82 indicates inaccuracies in the information presented in both

Table 2
Human evolution results for Lumos learning QG tool.

Metric	Percentage
Correctness	3.62
Relevance	4.01
Accuracy	2.82
Clarity	2.26
Depth of thought	2.04
Overall helpful	1.84

questions and answers. The clarity and depth of thought scores highlight issues in language presentation and the level of analysis, which impact the overall quality. The metric overall helpful at 1.84 suggests that the generated pairs lack practical usefulness and applicability, especially in educational contexts. Most of the questions are simple wh-questions with keyword-based or short answers, which contribute to the lower overall helpfulness score. For example:

- **Passage**: How about inflected forms like cats versus cat? These two words have the same lemma cat but are different wordforms. A lemma is a set of lexical forms having the same stem, the same major part-of-speech, and the same word sense. The word-form is the full inflected or derived form of the word. For morphologically complex languages like Arabic, we often need to deal with lemmatization.
  Generated Question: What is the name of a lemma word?
  Generated Answer: Cat
  In this example, the generated QA pair are not relevant to the context scope, as the passage does not discuss the specific names of lemma words. The question itself is unclear, and the answer does not reflect a meaningful understanding of the question. This highlights a limitation in the tool's ability to analyse the context and generate accurate QA pairs that reflect the passage content.
- **Passage**: The goal of classification is to take a single observation, extract some useful features, and thereby classify the observation into one of a set of discrete classes. One method for classifying text is to use handwritten rules. There are many areas of language processing where handwritten rule-based classifiers constitute a state-of-the-art system, or at least part of it.
  Generated Question: What is the goal of classification?
  Generated Answer: A single observation.
  The answer in this example is incomplete. It lacks the necessary details to fully convey the comprehensive goal of classification described in the passage, which can be summarised as 'classifying the observation into one of a set of discrete classes'.
- **Passage**: If a chatbot is human-like, users are also more likely to disclose private information, and less likely to worry about the harm of this disclosure (Ischen et al. 2019). In general, chatbots that are trained on transcripts of human–human or human–machine conversation must anonymise personally identifiable information.
  Generated Question: Who is less likely to worry about the harm of a chatbot?
  Generated Answer: Ischen et al. 2019.
  The question generated is poorly constructed and lacks clarity and precision. The answer, 'Ischen et al. 2019', indicates that the tool had difficulty accurately analysing the reference in the passage. The reference in parentheses was mistakenly interpreted as being integral to the sentence's meaning, revealing a limitation in the tool's capability to handle citations. As a result of this misinterpretation, an incorrect QA pair was generated.

It is worth mentioning that the complex and technical nature of educational content, especially in scientific domains like data mining, which incorporate tables and equations, can influence the quality of the generated QA pairs. These complexities present challenges for automated tools, affecting their accuracy, clarity, and overall relevance. Recognising these challenges is crucial for a comprehensive understanding of the evaluation results and offers valuable insights for future advancements in tools generating QA pairs for educational purposes.

The exploration of developing educational chatbots within the traditional intent-based framework, specifically relying on QA datasets, reveals challenges in meeting the diverse and complex requirements of educational environments. The system needs to handle a range of student intents, from simple term enquiries to complex scientific explanations, with clear, accurate, and correct responses. Considering these challenges, our experiment moves towards the latest advancements in chatbot development. We examine the RAG approach, utilising LangChain and OpenAI GPT-3.5. This strategic shift aims to address previous limitations and create a more effective and adaptable educational chatbot. The following section outlines our methodology, results, and evaluation of experimenting with generative AI for educational assistance chatbots in higher education.

5.2. Evolution to large language model-based educational chatbot

In the process of evolving into an advanced educational chatbot, this section explores the integration of RAG using LangChain and the OpenAI GPT-3.5-Turbo model. LangChain is an open-source orchestration framework for developing applications using LLMs. LangChain's tools and APIs streamline the process of building LLM-driven chatbots using the RAG approach. The framework is constructed around two main concepts: components and use-case-specific chains. Components are modular abstractions and collections of implementations for each abstraction. These components include models, prompts, indexes, memory, chains, and agents. They are the individual elements that combine to create a complete system. On the other hand, use-case-specific chains are prebuilt chains designed to accomplish specific higher-level tasks. They can be used as a starting point to develop complex applications. The components can be used to create new chains or customise existing chains (LangChain, 2023). To fully capture the potential of LangChain, it is important to understand the role, capabilities, and requirements of each component.

- **Prompts**: This is essentially the main component for communicating with and guiding the behaviour of AI's LLMs. It consists of natural language text that incorporates the user's input (question) and instructions for the LLM on how to process the provided input in order to generate a desired response. A prompt may include a one-shot or a set of few-shot examples to direct the language model towards improved performance. LangChain offers several categories of specialised prompt templates for generating prompts for language models. A prompt template is a predefined method of creating a prompt for the model. It can include specific instructions for the language model, a few-shot example, and a given context along with the user's questions.
- **Models**: There are three main types of models used in LangChain: LLMs, chat models, and text embedding models. LLMs are the main class of models used in LangChain. These models take a text string as input and produce a text string as output. Chat models are a more structured API that works with Chat Messages as input and output. Finally, the text embedding model takes the input as text and converts it into a list of numbers (floats) to create a numerical representation of the provided text.
- **Chains**: In LangChain, the term 'chains' refers to the technique of creating a series of operations by combining LLMs with other components to form a unified application. These chains can connect one or multiple LLMs to a user prompt, allowing interaction with the provided text or other data in a specific manner. This approach is particularly useful for streamlining the development of intricate applications that involve chaining LLMs, either with each other or with other components. As depicted in Fig. 5, a simple chain may consist of a user input that is formatted using

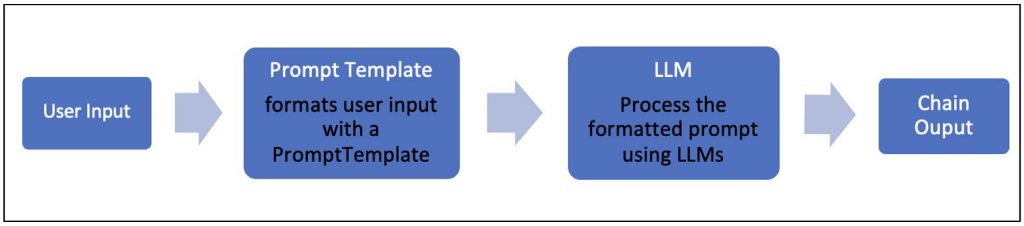

Fig. 5. Simple chain structure.

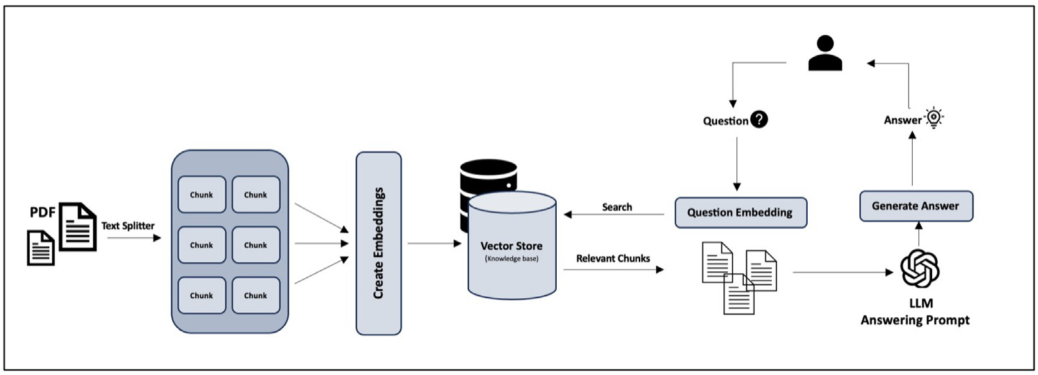

Fig. 6. Retrieval-augmented generation-based chatbot framework.

a specified prompt template and then processed using a language model to generate the desired output. For more complex tasks, multiple chains can be combined, with the output of one chain serving as the input for the next chain.

- **Memory**: In LangChain, the memory component enables LLM applications to store and retrieve information from previous interactions. The memory system is essential for developing a conversational interface ('chatbots') that can maintain context throughout a conversation and respond appropriately to follow-up questions. LLMs themselves are stateless, meaning that each request is handled independently of other interactions. Therefore, a memory component is required to provide LLMs with the user's previous requests, which will be processed alongside the user's new query.

5.2.1. Design and implementation

To build the chatbot, we used the OpenAI Python package to initialise OpenAI LLM classes. An OpenAI API key is required to access OpenAI APIs, which can be obtained by creating an OpenAI account. The development process, as shown in Fig. 6, involved the following tasks:

- **Load**: We used LangChain's Unstructured-PDFLoader to load our data, which included lecture transcripts and textbook chapters provided as PDFs.
- **Splitting Data to Chunks**: Typically, the context window of LLMs requires breaking long texts into smaller, semantically meaningful chunks that fit within the model's token limit. The context window of an LLM refers to the number of tokens it can accept as input to generate responses. Tokens can be characters, words, sentences, or other segments of text. To facilitate easier processing, the PDFs were segmented into smaller chunks using LangChain's CharacterTextSplitter.
- **Embeddings**: LLMs require text embeddings as a fundamental building component. These serve as a link between the neural network of the language model and the raw text input. Embeddings are numerical representations of words or phrases that capture their semantics and context, enabling the language model to understand the meaning and relationship of the given text in order to generate coherent and meaningful responses that mimic natural human communication. We utilised OpenAI Embeddings 'text-embedding-ada-002', which is recommended by OpenAI for its improved performance, cost-effectiveness, and ease of use. OpenAI Embeddings measure the relatedness of texts in various contexts for purposes such as text search, clustering, or classification. These embeddings are stored using a vector store, and for storing and retrieving embeddings, we used the Facebook AI Similarity Search (FAISS) Library. This library allows developers to quickly search for embeddings of multimedia documents that are similar to each other (Meta, 2023). When a user submits a question, the same model is used to embed the query, and relevant chunks with context are retrieved. Subsequently, the retrieved chunks are concatenated with the query and fed to the LLM chain to generate an answer.
- **Retrieval**: Upon receiving a question from a user, its embedding is calculated, and relevant segments are retrieved from the FAISS database using vector retrieval. These retrieved segments, along with the question, are then sent to the LLM to generate contextually appropriate answers to user queries.

Finally, a crucial element of LangChain is LLMs. Instead of acting as a server for its own LLMs, LangChain offers a standardised interface for interacting with multiple LLMs. For our experiment, we utilised the latest, most advanced, and cost-effective OpenAI language model, 'gpt-3.5-turbo'. Additionally, LangChain provides interfaces for both LLMs and Chat models. Chat models are a variant of language models that internally utilise language models, although their user interfaces have slight differences. They employ an interface where 'chat messages' serve as inputs and outputs, rather than a 'text in, text out' API. In this work, we constructed a chain using both models, along with other components, to develop an AI-powered chatbot.

To engage with the model, we must create a series of instructions or inputs that direct the model to generate the desired output. A prompt is essential in constructing LLM chains. It offers precise instructions to an LLM, along with context and queries that aid the model in comprehending the situation and generating appropriate and logical responses. In our model, two main prompts have been developed. The initial prompt is utilised to establish a ChatPromptTemplate, which defines the behavioural attributes of the chatbot. This guarantees that it operates within the given context and aligns with the intended persona of a teaching assistant.

```
Chat_PROMPT = """You are a teaching assistant chatbot.
Your task is to provide answers based on the provided
context and not from your own knowledge. Only answer user
questions that are directly related to the given context.
Context Information:
{context}
Your responses should be clear and directly relevant to
the context, and refrain from introducing new informa-
tion.
Now, please answer the user's question based on the con-
text:"""
```

The second PromptTemplate is designed to handle follow-up questions, preserving their content and context within the conversation. This prompt includes a chat history component that stores previous interactions, as well as a follow-up input that presents the original question. To capture the chat history, including previous questions and their corresponding answers, we utilise the ConversationBufferMemory module. This memory retrieval mechanism ensures that the user can easily refer back to the follow-up question during the conversation.

```
followUp_question_PROMPT = """Given the following con-
versation and a follow-up question, rephrase the follow-
up question to be a standalone question without changing
the content in the given question.
Chat History:
{chat_history}
Follow-Up Input: {question}
Standalone question:"""
```

Finally, the ConversationalRetrievalChain module is used to combine different elements, such as a vector retriever, a question generator, a document processing chain, and a memory component. This chain acts as a comprehensive module, coordinating the various LangChain components and models to generate intelligent and contextually appropriate answers to user queries.

5.2.2. Evaluation

To ensure a thorough evaluation, both automatic and human assessments were conducted. The performance of the chatbot was evaluated using various metrics, such as relevance, correctness, accuracy, clarity, depth of thought, and overall helpfulness. We used a set of 20 questions and answers that were manually generated from the course materials. For each question, the chatbot's response was compared to the correct answer to assess its overall performance.

Automatic evaluation. The automatic evaluation was conducted using the 'score evaluator' library provided by LangChain, which is used with OpenAI's GPT-4 model. The score evaluator instructs a language model to evaluate the model's prediction on a specific scale. In our evaluation, we used a 1–10 scoring scale, where higher scores indicate more accurate and relevant responses. Each question was evaluated using multiple metrics, including relevance, correctness, accuracy, clarity, depth of thought, and overall helpfulness. Table 3 presents an example of a question about the process of text normalisation, along with its reference answer compared to the predicted answer generated by the chatbot. The evaluation section provides system feedback on the generated answers, reflecting the chatbot's performance based on the specified metrics.

Table 3
Example 1 of automatic evaluation with GPT-4.

Question	What is the process of text normalisation?
Reference answer	Normalising text means converting it to a more convenient, standard form. The main steps in text normalisation are tokenisation, lemmatisation, stemming, sentence segmentation, and Parts-of-Speech (POS) tagging.
Predicted answer	`The process of text normalisation in natural language processing involves several tasks. These tasks include tokenising (segmenting) words, normalising word formats, and segmenting sentences. Tokenisation involves separating words from running text, taking into account cases where whitespace is not sufficient. Normalising word formats includes tasks like lemmatisation, which determines the common root of words, and stemming, which involves stripping suffixes from words. Sentence segmentation involves breaking up a text into individual sentences using punctuation cues.`
Evaluation	`The assistant's response is helpful and relevant to the user's question. It correctly and accurately explains the process of text normalisation in natural language processing, including tasks such as tokenising words, normalising word formats, and segmenting sentences. The assistant also provides an in-depth explanation of specific tasks like lemmatisation and stemming. However, the assistant failed to mention Parts-of-Speech (POS) tagging, which is a significant part of text normalisation.`
Score	8

The results of this automated evaluation show that the chatbot consistently performs well, with a significant proportion of its responses scoring 8 or higher. Two questions received a perfect score of 10, indicating exceptional performance across all evaluation metrics. Additionally, eight questions scored 9, demonstrating a very high level of correctness, relevance, accuracy, and overall helpfulness. A total of 7 out of 20 obtained a score of 8, and while these responses were notably correct, there is room for improvement in terms of depth and detail. Finally, one question garnered a score of 7, and two questions received a score of 6, signifying satisfactory performance with potential for improvement in certain aspects such as correctness, accuracy, and depth. For instance, as shown in Table 4, when examining the question regarding different approaches for extracting relations among entities in NLP, the assistant's response received a score of 6. The feedback indicated that while the response correctly included supervised ML, it lacked coverage of various other methods such as pattern-based approaches, lightly supervised bootstrapping methods, distant supervision, and unsupervised or Open IE methods, which were part of the ground truth.

Human evaluation. In addition to automated assessment, the responses and overall performance of the chatbot were critically reviewed and evaluated by two annotators who specialise in the field of Data Mining and Text Analytics. This thorough assessment took into account the chatbot's contextual relevance and understanding of the specific domain. Each response from the chatbot was evaluated using the same metrics as those used in the automated evaluation, including relevance, correctness, accuracy, clarity, depth of thought, and overall helpfulness. Each metric was measured on a scale of 1 to 5, where lower scores indicated poorer performance in the corresponding metric, and higher scores indicated higher-quality results.

The results, as summarised in Table 5, demonstrated a very satisfactory level of correctness, relevance, and clarity, achieving scores of 4.7, 4.9, and 4.78, respectively. These findings highlight the chatbot's ability to provide both grammatically and semantically accurate and relevant answers to students' enquiries. The accuracy metric achieved a score of 4.63, while the depth of thought and overall helpfulness received scores of 4.65 and 4.88, respectively. This suggests that although most of the chatbot's responses were understandable and on-topic,

Table 4
Example 2 of automatic evaluation with GPT-4.

Question	What are the different approaches for extracting relations among entities in natural language processing?
Reference answer	Relations among entities can be extracted by pattern-based approaches, supervised learning methods when annotated training data is available, lightly supervised bootstrapping methods when small numbers of seed tuples or seed patterns are available, distant supervision when a database of relations is available, and unsupervised or Open IE methods.
Predicted answer	`The different approaches for extracting relations among entities in natural language processing include supervised machine learning, feature-based classification, and neural classification. These approaches involve annotating a training corpus with relations and entities, and then using classifiers to annotate unseen test sets.`
Evaluation	`The assistant's response is partially correct and relevant. It includes supervised machine learning which is one of the approaches mentioned in the ground truth. However, it does not include all the other methods such as pattern-based approaches, lightly supervised bootstrapping methods, distant supervision, and unsupervised or Open IE methods. The depth of the response is limited as it does not provide a comprehensive answer to the question. The assistant's response could be more helpful if it included all the approaches mentioned in the ground truth and provided a brief explanation of each.`
Score	6

Table 5
Human evaluation results for retrieval-augmented generation–based chatbot.

Metric	Percentage
Correctness	4.7
Relevance	4.9
Accuracy	4.63
Clarity	4.78
Depth of thought	4.65
Overall helpful	4.88
Overall Average ≈ 4.76	

there is still room for improvement in terms of accuracy and depth of thought to better meet the specific needs of student queries. Nonetheless, the high overall average score of approximately 4.76 indicates the effectiveness of the chatbot in delivering reliable and valuable educational information.

6. Discussion and future directions

There are significant limitations of traditional intent-based chatbots compared to LLM-based chatbots. In an educational context, the challenges faced during manual intent creation for the Alexa skill highlight the unique requirements of adapting chatbots for learning environments. Unlike task-oriented chatbots, where the conversation can be guided and predicted, educational chatbots encounter greater complexity as the dialogue is driven by diverse student needs, ranging from simple to complex enquiries. Traditional chatbots rely on matching user requests to predefined intents and sample utterances. Their dependence on predefined datasets limits their ability to handle complex or open-ended queries, whereas LLMs have much broader language understanding capabilities. This capability is particularly valuable in educational settings where students seek detailed explanations and guidance.

Table 6
Example of complex question.

Question	What are the fundamental differences between generative classifiers like naive Bayes and discriminative classifiers like logistic regression in the context of machine learning for text classification?
Predicted answer	`In text classification, generative classifiers and discriminative classifiers differ in their approaches. Generative classifiers like naive Bayes build a model of how a class could generate some input data. Given an observation, they return the class most likely to have generated the observation. Discriminative classifiers like logistic regression instead learn what features from the input are most useful to discriminate between the different possible classes. While discriminative systems are often more accurate and hence more commonly used, generative classifiers still have a role.`

Furthermore, the experiment with Alexa highlights the time-consuming process of manually creating intents, specifying relevant utterances, and generating appropriate responses for educational chatbots. This inefficiency can have a significant impact on chatbot performance, especially in situations where scalability and efficiency are crucial. Additionally, the efficiency of traditional chatbot responses is often influenced by the quality of its knowledge base. Creating a comprehensive question–answer dataset for educational content is a challenging task. It involves not only capturing the depth of information from the course but also aims to simulate the proficient understanding and expertise of a teacher persona. While the Lumos Learning QA tool successfully generates a massive dataset of 9000 QA pairs, it emphasises the critical importance of high-quality educational content. In the academic domain, the need for accurate, contextually relevant, and well-structured question–answer pairs is particularly important. However, the evaluation results highlight that the majority of the generated QA are simple wh-questions, which do not meet the required standards. It is essential to emphasise that the generated questions should cover different question types, not just simple wh-questions, to build a powerful educational knowledge base that can effectively address different student enquiries.

In contrast, the RAG approach addresses these challenges by combining retrieval and generation algorithms. By incorporating educational materials, the system directly retrieves relevant passages in response to student questions and generates appropriate responses based on those queries. Integrating generative AI chatbots into the field of education has the potential to enhance the student learning experience. The evaluation, both automatic and human, demonstrates the significant potential of generative AI chatbots in the role of teaching assistants. The proposed system exhibits a highly satisfactory level in handling student questions, ranging from simple wh-questions to more complex queries. For instance, Table 6 presents an evaluated question that is considered relatively complex, as it requires a deep understanding of both generative and discriminative classifiers, their underlying principles, and how they operate in the specific context of text classification. The generated answer provides a clear and concise explanation of the fundamental differences between both classifiers, reflecting the chatbot's ability to comprehend complex queries. In education, utilising such tools will provide students with access to a vital support system that is available around the clock and offers direct, personalised responses.

Moreover, by using conversational memory, the proposed RAG system successfully handled follow-up questions, which is essential for

Table 7
Comparison of intent-based vs. retrieval-augmented generation approaches.

Aspect	Intent-based approach	Retrieval-augmented generation approach
Foundation	Primarily relies on predefined intents and corresponding responses, limiting the scope of educational content.	Combines retrieval and generation mechanisms, allowing access to a broader range of educational content.
Flexibility	Limited flexibility as responses are based on predefined intents, which may not address all educational topics.	Highly flexible, adapts to a wide range of student queries through retrieval, handling various educational topics.
Adaptability	Struggles with handling queries outside predefined intents, potentially not considering complex educational queries.	Adaptable to diverse educational queries by retrieving and augmenting relevant information, providing comprehensive answers.
User interaction	Students need to conform to predefined intents for effective interaction, which may limit their engagement and exploration.	Allows more natural and context-aware interactions, encouraging deeper student engagement with educational material.
Knowledge base	Requires a well-defined set of intents and associated responses, which may become outdated and limited in educational scope.	Knowledge base can be expanded by retrieving information as needed, ensuring that educational content is up-to-date and comprehensive.
Development complexity	Generally simpler development due to predefined intents, facilitating the development process but resulting in less dynamic educational applications.	May require more complex development, especially in designing prompts and integrating retrieval, but offers richer educational interactions.
Handling new information	Challenges in handling queries that do not align with predefined intents, limiting the ability to address new or emerging educational queries.	More proficient at handling novel queries by retrieving and generating relevant educational information.
Use of artificial intelligence models	May involve AI models for intent recognition and response mapping, which could be limited in educational adaptability.	Integrates AI models for both retrieval and generation, combining their strengths to offer more robust educational responses.
Scalability	Scalability may be limited by the need for constant intent expansion, which can be resource-intensive and slow to adapt to new educational content.	Potentially more scalable, as new educational content can be much more easily embedded.
Context awareness	Limited context awareness as responses are typically based on current intent, which can limit the ability to provide context-rich educational experiences.	Improved context awareness as responses can be augmented by retrieved information, facilitating deeper and more context-rich educational interactions.

engaging students in meaningful conversations. However, in lengthy and complex dialogues, the conversation context might exceed the ‘token limit’ of the LLM. The context window represents the amount of text the LLM can process when generating responses, including system prompts, previous conversations, and individual queries. This limitation could potentially hinder the model’s ability to sustain long and coherent conversations. To ensure optimal performance of the chatbot, it is crucial to consider the model’s context window and its associated costs during the development of AI chatbots. In our case, we chose GPT-3.5-Turbo due to its cost-effectiveness and efficiency compared to GPT-4.

Table 7 presents a comparison between the intent-based and RAG approaches, outlining key differences in the development of educational chatbots. In the educational context, the intent-based approach relies on predefined intents, which limits both flexibility and adaptability in responding to user queries. Students must adhere to predetermined intents, and the system may struggle with queries outside of this scope. In contrast, the RAG approach with LLMs combines retrieval and generation mechanisms, offering greater flexibility and adaptability to diverse educational queries. This approach allows for more natural interactions, excels in handling new and complex queries, and benefits from an expandable knowledge base through dynamic retrieval, which is crucial for educational environments that require up-to-date and diverse information. While the intent-based approach involves simpler development, the RAG approach may involve more complexity, especially in designing prompts and integrating retrieval processes. Overall, the RAG approach with LLMs demonstrates enhanced scalability, context awareness, and proficiency in handling a wider range of user queries, making it a promising framework for the development of educational chatbots.

In the educational setting, user experiences are crucial in determining the effectiveness of learning tools. Therefore, future research will focus on obtaining user feedback by deploying the chatbot for students enrolled in the data mining and text analytics course at the University of Leeds. This practical application will facilitate further testing and evaluation, providing valuable insights into how the chatbot impacts students’ learning experiences in the real world.

7. Conclusion

This research investigates the evolution of educational chatbots from traditional intent-based models to the advanced RAG technique. It explores different approaches to developing a teaching assistant chatbot for Data Mining and Text Analytics courses at the University of Leeds. The findings reveal limitations in the intent-based approach created using Alexa skills, such as limited flexibility and difficulties in handling queries outside of predefined intents. Furthermore, the traditional approach often required a well-defined set of QA pairs that reflected user intents. In contrast to manual approaches that are time-consuming, automated tools like the Lumos Learning QA show promise in generating large QA datasets. However, evaluation results indicate issues with correctness, clarity, accuracy, and overall helpfulness in the majority of generated QAs. To address these challenges, the research proposes an RAG-based teaching chatbot that combines retrieval and generative methodologies. Implemented using the LangChain framework with the OpenAI GPT-3.5 Turbo model, both automatic and human evaluations indicate a high level of performance in efficiently handling a wide range of student questions. This shift represents a significant advancement in the capabilities of educational chatbots, revolutionising the way students access information and engage with learning resources.

CRediT authorship contribution statement

Bashaer Alsafari: Writing – review & editing, Writing – original draft, Visualization, Software, Resources, Project administration, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Eric Atwell:** Writing – review & editing, Visualization, Validation, Supervision, Resources, Project administration, Methodology, Data curation, Conceptualization. **Aisha Walker:** Visualization, Validation, Supervision, Project administration, Methodology, Conceptualization. **Martin Callaghan:** Writing – review & editing, Visualization, Validation, Supervision, Project administration, Methodology, Investigation, Formal analysis, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

We extend our acknowledgement and appreciation to King Abdulaziz University and the University of Leeds, for their continuous support throughout our research. We also extend our acknowledgement and appreciation to Dan Jurafsky and James Martin for making available to the NLP community their draft text for the third edition of their textbook "Speech and Language Processing", recommended reading for Data Mining and Text Analytics students: https://web.stanford.edu/~jurafsky/slp3/.

Appendix. Data mining and text analytics lecture transcripts

Lecture transcripts included in the corpus:

- Text pre-processing transcript.
- Data mining and text analytics online communities and applications transcript.
- Text classification transcript.
- N-grams transcript.
- Tagging POS and NER transcript.
- Scaling to big data transcript.
- Word meanings transcript.
- Machine Translation: Challenges and Approaches transcript.
- Information Extraction: Extracting Named Entities and Relations from Text transcript.
- Cheat, NLTK, SpaCy: Text Analytics in Python transcript.
- Coursework report: produce an applied text analytics research proposal transcript.
- Chatbots and Dialogue Systems transcript.
- University student surveys using chatbots: Artificial Intelligence conversational agents transcript.
- Information retrieval and web search transcript.
- Multi-word expressions transcript.
- Deep learning of text understanding: Google's BERT transcript.

References

Aithal, S.G., Rao, A.B., Singh, S., 2021. Automatic question-answer pairs generation and question similarity mechanism in question answering system. Appl. Intell. 1–14.

Alexa, A., 2023a. Custom skills. https://developer.amazon.com/en-US/alexa/alexa-skills-kit/get-deeper/custom-skills. (Accessed 2 July 2023).

Alexa, A., 2023b. Developer guide: What is aws lambda? https://docs.aws.amazon.com/lambda/latest/dg/welcome. (Accessed 5 July 2023).

Alexa, A., 2023c. What is alexa? https://developer.amazon.com/en-US/alexa. (Accessed 1 July 2023).

Alexa, A., 2024. Voice design guide:What users say https://developer.amazon.com/fr/designing-for-voice/what-users-say/#handle-over-answering. (Accessed 22 August 2024).

Alnefaie, S., Atwell, E., Alsalka, M.A., 2023. Using automatic question generation web services tools to build a quran question-and-answer dataset. Int. J. Islam. Appl. Comput. Sci. Technol. 11, 1–12.

Altinok, D., 2018. An ontology-based dialogue management system for banking and finance dialogue systems. arXiv preprint arXiv:1804.04838.

Anil, R., Dai, A.M., Firat, O., Johnson, M., Lepikhin, D., Passos, A., Shakeri, S., Taropa, E., Bailey, P., Chen, Z., et al., 2023. Palm 2 technical report. arXiv preprint arXiv:2305.10403.

Assayed, S., Shaalan, K., Alkhatib, M., 2023. A chatbot intent classifier for supporting high school students. EAI Endorsed Trans. Scalable Inf. Syst. 1.

Black, S., Biderman, S., Hallahan, E., Anthony, Q., Gao, L., Golding, L., He, H., Leahy, C., McDonell, K., Phang, J., 2022. Gpt-neox-20b: An open-source autoregressive language model. arXiv preprint arXiv:2204.06745.

Carayannopoulos, S., 2018. Using chatbots to aid transition. Int. J. Inf. Learn. Technol. 35, 118–129.

Chen, H., Liu, X., Yin, D., Tang, J., 2017. A survey on dialogue systems: Recent advances and new frontiers. ACM SIGKDD Explor. Newsl. 19, 25–35.

Chien, Y.H., Yao, C.K., 2020. Development of an ai userbot for engineering design education using an intent and flow combined framework. Appl. Sci. 10 (7970).

Clarizia, F., Colace, F., Lombardi, M., Pascale, F., Santaniello, D., 2018. Chatbot: An education support system for student. In: Cyberspace Safety and Security: 10th International Symposium, CSS 2018, Amalfi, Italy, October, 2018 29–31, Proceedings 10. Springer, pp. 291–302.

Cui, S., Bao, X., Zu, X., Guo, Y., Zhao, Z., Zhang, J., Chen, H., 2021. Onestop qamaker: Extract question-answer pairs from text in a one-stop approach. arXiv preprint arXiv:2102.12128.

El Alaoui, H., El Aouene, Z., Cavalli-Sforza, V., 2023. Building intelligent chatbots: Tools,technologies, and approaches. In: 2023 3rd International Conference on Innovative Research in Applied Science, Engineering and Technology. IRASET, IEEE, pp. 1–12.

Eom, S.B., Ashill, N., 2016. The determinants of students' perceived learning outcomes and satisfaction in university online education: An update. Decision Sciences Journal of Innovative Education 14, 185–215.

Folstad, A., Araujo, T., Law, E.L.C., Brandtzaeg, P.B., Papadopoulos, S., Reis, L., Baez, M., Laban, G., McAllister, P., Ischen, C., 2021. Future directions for chatbot research: An interdisciplinary research agenda. Computing 103, 2915–2942.

Franco, M.F., Rodrigues, B., Scheid, E.J., Jacobs, A., Killer, C., Granville, L.Z., Stiller, B., 2020. Secbot: A business-driven conversational agent for cybersecurity planning and management. In: 2020 16th International Conference on Network and Service Management. CNSM, IEEE, pp. 1–7.

Fryer, L.K., Ainley, M., Thompson, A., Gibson, A., Sherlock, Z., 2017. Stimulating and sustaining interest in a language course: An experimental comparison of chatbot and human task partners. Comput. Hum. Behav. 75, 461–468.

Hien, H.T., Cuong, P.N., Nam, L.N.H., Nhung, H.L.T.K., Thang, L.D., 2018. Intelligent assistants in higher-education environments: The fit-ebot, a chatbot for administrative and learning support. In: Proceedings of the 9th International Symposium on Information and Communication Technology. pp. 69–76.

Hobert, S., 2019. Say Hello to 'Coding Tutor'! Design and Evaluation of a Chatbot-Based Learning System Supporting Students to Learn to Program.

Jawahar, G., Sagot, B., Seddah, D., 2019. What does bert learn about the structure of language? In: ACL 2019-57th Annual Meeting of the Association for Computational Linguistics.

Jia, J., 2009. Csiec: A computer assisted english learning chatbot based on textual knowledge and reasoning. Knowl.-Based Syst. 22, 249–255.

Khadija, M.A., Aziz, A., Nurharjadmo, W., 2023. Automating information retrieval from faculty guidelines: Designing a pdf-driven chatbot powered by openai chatgpt. In: 2023 International Conference on Computer, Control, Informatics and Its Applications, IC3INA. IEEE, pp. 394–399.

LangChain, 2023. Langchain documentations:introduction. https://python.langchain.com/docs/get_started/introduction. (Accessed 25 September 2023).

Lee, Y.C., Yamashita, N., Huang, Y., 2021. Exploring the effects of incorporating human experts to deliver journaling guidance through a chatbot. Proc. ACM Hum.-Comput. Interact. 5, 1–27.

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.t., Rocktäschel, T., 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. Adv. Neural Inf. Process. Syst. 33, 9459–9474.

Liu, R., Zenke, C., Liu, C., Holmes, A., Thornton, P., Malan, D.J., 2024. Teaching cs50 with ai: Leveraging generative artificial intelligence in computer science education. In: Proceedings of the 55th ACM Technical Symposium on Computer Science Education V.1. pp. 750–756.

Luo, B., Lau, R., Li, C., Si, Y., 2022. A critical review of state-of-the-art chatbot designs and applications. WIREs Data Min. Knowl. Discov. 12 (1), e1434.

Maryamah, M., Irfani, M.M., Tri Raharjo, E.B., Rahmi, N.A., Ghani, M., Raharjana, I.K., 2024. Chatbots in academia: A retrieval-augmented generation approach for improved efficient information access. In: 2024 16th International Conference on Knowledge and Smart Technology. KST, pp. 259–264. http://dx.doi.org/10.1109/KST61284.2024.10499652.

Meta, 2023. Faiss-facebook ai similarity search. https://ai.meta.com/tools/faiss/. (Accessed 4 October 2023).

Meyer vonWolff, R., Hobert, S., Schumann, M., 2021. Sorry, i can't understand you!–influencing factors and challenges of chatbots at digital workplaces. In: Innovation Through Information Systems: Volume II: A Collection of Latest Research on Technology Issues. Springer, pp. 150–165.

Misargopoulos, A., Nikolopoulos-Gkamatsis, F., Nestorakis, K., Tzoumas, A., Giannakopoulos, G., Gizelis, C.A., Kefalogiannis, M., 2022. Building a knowledge-intensive, intent-lean, question answering chatbot in the telecom industry-challenges and solutions. In: IFIP International Conference on Artificial Intelligence Applications and Innovations. Springer, pp. 87–97.

Neupane, S., Hossain, E., Keith, J., Tripathi, H., Ghiasi, F., Golilarz, N.A., Amirlatifi, A., Mittal, S., Rahimi, S., 2024. From questions to insightful answers: Building an informed chatbot for university resources. arXiv preprint arXiv:2405.08120.

Okonkwo, C.W., Ade-Ibijola, A., 2020. Python-bot: A chatbot for teaching python programming. Engineering Letters 29.

Perez, J.Q., Daradoumis, T., Puig, J.M.M., 2020. Rediscovering the use of chatbots in education: A systematic literature review. Comput. Appl. Eng. Educ. 28, 1549–1565.

Pham, X.L., Pham, T., Nguyen, Q.M., Nguyen, T.H., Cao, T.T.H., 2018. Chatbot as an intelligent personal assistant for mobile language learning. In: Proceedings of the 2018 2nd International Conference on Education and E-Learning. pp. 16–21.

Rahman, A., Al Mamun, A., Islam, A., 2017. Programming challenges of chatbot: Current and future prospective. In: 2017 IEEE Region 10 Humanitarian Technology Conference (R10-HTC). IEEE, pp. 75–78.

Ray, P.P., 2023. Chatgpt: A comprehensive review on background, applications, key challenges, bias, ethics, limitations and future scope. Internet of Things and Cyber-Physical Systems.

Rosruen, N., Samanchuen, T., 2018. Chatbot utilization for medical consultant system. In: 2018 3rd Technology Innovation Management and Engineering Science International Conference. TIMES-ICON, IEEE, pp. 1–5.

Ruan, S., Willis, A., Xu, Q., Davis, G.M., Jiang, L., Brunskill, E., Landay, J.A., 2019. Bookbuddy: Turning digital materials into interactive foreign language lessons through a voice chatbot. In: Proceedings of the Sixth, 2019 ACM Conference on Learning@ Scale. pp. 1–4.

Shawar, Bayan Abu, Atwell, Eric, 2007. Chatbots: are they really useful?. Journal for Language Technology and Computational Linguistics 22 (1), 29–49.

Song, D., Oh, E.Y., Rice, M., 2017. Interacting with a conversational agent system for educational purposes in online courses. In: 2017 10th International Conference on Human System Interactions. HSI, IEEE, pp. 78–82.

Sophia, J.J., Jacob, T.P., 2021. Edubot-a chatbot for education in covid-19 pandemic and vqabot comparison. In: 2021 Second International Conference on Electronics and Sustainable Communication Systems. ICESC, IEEE, pp. 1707–1714.

Soygazi, F., Oguz, D., 2023. An analysis of large language models and langchain in mathematics education. In: Proceedings of the 2023 7th International Conference on Advances in Artificial Intelligence. pp. 92–97.

Spector, J.M., 2014. Conceptualizing the emerging field of smart learning environments. Smart Learn. Environ. 1, 1–10.

Thoppilan, R., De Freitas, D., Hall, J., Shazeer, N., Kulshreshtha, A., Cheng, H.T., Jin, A., Bos, T., Baker, L., Du, Y., et al., 2022. Lamda: Language models for dialog applications. arXiv preprint arXiv:2201.08239.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al., 2023. Llama 2: Open foundation and fine-tuned chat models. arXiv preprint arXiv:2307.09288.

Wang, Y., Liu, S., Song, L., 2022. Designing an educational chatbot with joint intent classification and slot filling. In: 2022 IEEE International Conference on Teaching, Assessment and Learning for Engineering. TALE, IEEE, pp. 381–388.

Willis, A., Davis, G., Ruan, S., Manoharan, L., Landay, J., Brunskill, E., 2019. Key phrase extraction for generating educational question-answer pairs. In: Proceedings of the Sixth, 2019 ACM Conference on Learning@ Scale. pp. 1–10.

Zhao, W.X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., Min, Y., Zhang, B., Zhang, J., Dong, Z., 2023. A survey of large language models. arXiv preprint arXiv:2303.18223.